

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method, comprising:
analyzing a first code; and
generating a second code as a single program based on the first code, the second code including
a microarchitecture implementation-specific representation of a portion of the first code,
and
a macroinstruction representation of the portion of the first code.
2. (Original) The method of claim 1, wherein the first code comprises source code.
3. (Original) The method of claim 1, wherein the first code comprises code compiled from source code for a given instruction set architecture (ISA).
4. (Cancelled)
5. (Currently Amended) The method of claim 4, further comprising generating boundary markers to mark a beginning and an end for the microarchitecture implementation-specific ~~alternative~~ representation, the boundary markers being in a format of the ISA.
6. (Currently Amended) The method of claim 5, further comprising generating a trigger instruction which when executed by a machine executing the second code causes the machine to execute the microarchitecture implementation-specific ~~alternative~~ representation instead of the macroinstruction representation ~~first code~~, the trigger being in the format of the ISA.
7. (Cancelled)
8. (Previously Presented) A machine-readable medium having stored thereon a sequence of instructions which when executed by a processor, cause the processor to perform a set of operations comprising:
generating compiled code for a given ISA , the compiled code including a)
discrete regions of microarchitecture implementation-specific code bounded by ISA format markers and
b) macroinstructions outside the discrete regions.

9. (Original) The machine-readable medium of claim 8, wherein the microarchitecture implementation-specific code comprises an alternative representation of the compiled code.
10. (Original) The machine-readable medium of claim 9, wherein the code sequence further comprises a trigger instruction in the format of the ISA which when executed by hardware causes the microarchitecture implementation-specific code to be executed instead of the compiled code for the ISA.
11. (Currently Amended) A microprocessor, comprising:
a fetch unit; and
a decode unit, the microprocessor having first and second modes of operation, wherein in the first mode, the decode unit decodes ISA format instructions supplied by the fetch unit, and in the second mode, the decode unit processes microarchitecture implementation-specific format instructions supplied by the fetch unit and the microprocessor transitions between the first and second modes upon detection by the decode unit of an ISA format boundary marker supplied by the fetch unit, both the ISA format instructions and the microarchitecture implementation-specific format instructions are within a same sequence of instructions when supplied by the fetch unit.
12. (Cancelled)
13. (Original) The microprocessor of claim 11, wherein processing the microarchitecture implementation-specific format instruction comprises decoding the instruction.
14. (Currently Amended) A method, comprising:
analyzing a first code comprising instructions for a first ISA; and
generating a second code as a single program based on the first code, the second code including at least some instructions for a second ISA corresponding to instructions in the first code, the second code further comprising microinstructions and macroinstructions.
15. (Cancelled)

16. (Original) The method of claim 14, wherein the first code is for execution by a first processing unit; and the second code is for execution by a second processing unit which supports a different ISA from the first processing unit.

17. (Currently Amended) A machine-readable medium having stored thereon a sequence of instructions which when executed by a processor, cause the processor to perform a method comprising:

analyzing a first code; and

generating a second code as a single program based on the first code, the second code including

a microarchitecture implementation-specific representation of a portion of the first code, and

a macroinstruction representation of the portion of the first code.

18. (Original) The machine-readable medium of claim 17, wherein the first code comprises source code.

19. (Original) The machine-readable medium of claim 17, wherein the first code comprises code compiled from source code for a given ISA.

20. (Currently Amended) A system, comprising:

a memory; and

a microprocessor coupled to the memory, the microprocessor including a fetch unit and a decode unit, wherein the microprocessor has first and second modes of operation, wherein in the first mode, the decode unit decodes ISA format instructions supplied by the fetch unit, and in the second mode, the decode unit processes microarchitecture implementation-specific format instructions supplied by the fetch unit and the microprocessor transitions between the first and second modes upon detection by the decode unit of an ISA format boundary marker supplied by the fetch unit, both the ISA format instructions and the microarchitecture implementation-specific format instructions are within a same sequence of instructions when supplied by the fetch unit.

21. (Cancelled)

22. (Currently Amended) A computer-readable medium having stored thereon a sequence of instructions which when executed by the processor, cause the processor to perform a method comprising:

analyzing a first code comprising instructions for a first ISA; and

generating a second code as a single program based on the first code, the second code including at least some instructions for a second ISA format corresponding to instructions in the first code, the second code further comprising microinstructions and macroinstructions.

23. (Cancelled)

24. (Original) The system of claim 22, wherein the first code is for execution by a first processing unit; and the second code is for execution by a second processing unit which support a different ISA from the first processing unit.

25. (Currently Amended) A system, comprising:

a processor; and

a memory coupled to the processor, the memory storing instructions which are executed by the processor, cause the processor to perform a method comprising;

analyzing a first code; and

generating a second code as a single program based on the first code, the second code including

a microarchitecture implementation-specific representation of a portion of the first code, and

a macroinstruction representation of the portion of the first code.

26. (Original) The system of claim 25, wherein the first code comprises source code.

27. (Original) The system of claim 26, wherein the first code comprises code compiled from source code from a given ISA.

28. (Currently Amended) A system, comprising:
a processor; and
a memory coupled to the processor, the memory storing instructions which are executed by the processor cause the processor to perform a method comprising;
analyzing the first code comprising instructions for a first ISA;
generating a second code as a single program based on the first code, the second code including at least some instructions for a second ISA format corresponding to instructions in the first code, the second code further comprising microinstructions and macroinstructions.
29. (Cancelled)
30. (New) A method, comprising:
loading a source program;
compiling the source program into a first compiled code including macroinstructions for an instruction set architecture (ISA);
optimizing the first compiled code into a second compiled code, the second compiled code including a macroinstruction segment and microinstruction segment, the microinstruction segment representing an alternate implementation of a function of the macroinstruction segment;
storing the second compiled code as a single compiled program.
31. (New) The method of claim 30, further comprising:
generating boundary markers to mark a beginning and an end of the microinstruction segment, the boundary markers being in a format of the ISA.